Amendments to the Claims:

1. (Currently Amended) A method for performing technology mapping, the method comprising:

a) receiving a design that is not bounded to a particular technology;

b) repeatedly:

selecting from the design a candidate sub-network[[,]];

generating a parameter based on a set of output Boolean functions performed by the selected candidate sub-network[[,]];

based on the parameter, identifying at least one replacement sub-network from a storage structure that stores replacement sub-networks that are bound to the particular technology[[,]]; and

replacing the selected candidate sub-network in the design with the replacement sub-network,

~~c)~~ wherein at least some of the selected candidate sub-networks have a graph structure comprising a first node having a first output outside the graph structure and a second node having a second output outside the graph structure, wherein the first node receives a direct or indirect input from the second node ~~that is different from a tree structure or a micro-leaf directed acyclic graph structure~~.

2. (Original) The method of claim 1 further comprising:

using the parameter to retrieve the replacement sub-network from the storage structure.

3. (Canceled)

4. (Currently Amended) The method of claim 1, wherein the set of output Boolean functions includes ~~one~~ two or more output Boolean functions.

5. (Original) The method of claim 1 further comprising:

terminating the repetitions once a stopping criteria is reached.

6. (Currently Amended) The method of claim 5, wherein the design includes a plurality of circuit elements and the sub-networks are formed by circuit elements, the method further comprising:

after terminating the repetitions, traversing the design to identify circuit elements that are not bound to the particular technology ~~library~~;

for each identified circuit element, attempting to identify a replacement sub-network that is stored in the storage structure; and

if at least one replacement sub-network for an identified circuit element is identified, replacing the circuit element in the design with the identified replacement sub-network.

7. (Currently Amended) The method of claim 6, wherein if more than one replacement ~~sub-networks are~~ sub-network is identified for a circuit element, selecting one of the replacement sub-networks and replacing the circuit element with the selected replacement sub-network.

8. (Currently Amended) The method of claim 7, wherein if more than one replacement ~~sub-networks are~~ sub-network is identified for a circuit element, selecting the replacement sub-network that is better than or as good as the rest of the identified replacement sub-networks.

9. (Original) The method of claim 6, wherein each circuit element performs a function, wherein if no replacement sub-network is identified for an identified circuit element, decomposing the function of the circuit element into a set of functions, and then attempting to identify a set of replacement sub-networks in the storage structure that perform the set of functions.

10. (Original) The method of claim 6, wherein traversing the design to identify circuit elements comprises identifying circuit elements that existed in the design when the design was received.

11. (Currently Amended) The method of claim 6 further comprising:

after traversing the design repeatedly:

selecting from the design a candidate sub-network;

identifying at least one replacement sub-network from a storage structure that stores replacement sub-networks; and

replacing the selected candidate sub-network in the design with the replacement sub-network.

12. (Currently Amended) The method of claim 1 further comprising:

before replacing ~~the~~ a candidate ~~sub-networks~~ sub-network with ~~the~~ a replacement ~~sub-networks~~ sub-network, evaluating whether to replace the selected candidate sub-network with the replacement sub-network,

wherein, during the repetitions, certain candidate sub-networks are replaced by replacement sub-networks based on the evaluations ~~evaluation, wherein~~ and certain candidate sub-networks are not replaced based on the evaluations.

13. (Currently Amended) The method of claim ~~13~~ 12, wherein the evaluating comprises computing a cost function.

14. (Currently Amended) A computer program embedded on a computer readable medium, the computer program for receiving a design that is not bounded to a particular technology and for mapping the design to the particular technology, the computer program comprising:

a first set of instructions for selecting from the design a candidate sub-network [[,]];

a second set of instructions for generating a parameter based on a set of output Boolean functions performed by the selected candidate sub-network [[,]];

a third set of instructions ~~fro~~ for identifying, based on the parameter, at least one replacement sub-network from a storage structure that stores replacement sub-networks that are bound to the particular technology[[,]];

a fourth set of instructions for replacing the selected candidate sub-network in the design with the replacement sub-network[[,]]; and

a fifth set of instructions for repeatedly executing the first to fourth sets of instructions,

wherein at least some of the selected candidate sub-networks have a graph structure comprising a first node having a first output outside the graph structure and a second node having a second output outside the graph structure, wherein the first node receives a direct or indirect input from the second node ~~that is different from a tree structure or a micro-leaf directed acyclic graph structure~~.

15. (Original) The computer program of claim 14, wherein the fifth set of instructions determines whether the computer program has reached a criterion for stopping the repetitions, wherein when the fifth set of instructions terminates the repetitions once the stopping criterion is reached.

16. (Currently Amended) The computer program of claim 15, wherein the design includes a plurality of circuit elements and the sub-networks are formed by circuit elements, the computer program further comprising:

a sixth set of instructions for traversing the design, after terminating the repetitions, to identify circuit elements that are not bound to the particular technology ~~library~~[[,]];

a seventh set of instructions for attempting to identify, for each identified circuit element, a replacement sub-network that is stored in the storage structure[[,]]; and

an eight set of instructions for replacing a circuit element when at least one replacement sub-network is identified for the circuit element.

17. (Currently Amended) The computer program of claim 16, wherein each circuit element performs a function, the computer program further comprising:

a ninth set of instructions for decomposing the function of a circuit element into a set of functions when no replacement sub-network is identified for a circuit element; and

a tenth set of instructions for attempting to identify a set of replacement sub-networks in the storage structure that perform the set of functions.

18. (New) The method of claim 1, wherein the design includes a plurality of circuit elements, the candidate sub-networks are formed by circuit elements, and a circuit element is represented by a node in a graph structure of a candidate sub-network.

19. (New) The method of claim 1, wherein a graph structure comprises a plurality of nodes at different levels, the first node is a top-level node that receives direct or indirect inputs from each node of the graph structure, and the second node is a lower-level node that does not receive direct or indirect inputs from each node of the graph structure.

20. (New) The computer program of claim 14, wherein the design includes a plurality of circuit elements, the candidate sub-networks are formed by circuit elements, and a circuit element is represented by a node in a graph structure of a candidate sub-network.

21. (New) The computer program of claim 14, wherein a graph structure comprises a plurality of nodes at different levels, the first node is a top-level node that receives direct or indirect inputs from each node of the graph structure, and the second node is a lower-level node that does not receive direct or indirect inputs from each node of the graph structure.